**\_\_\_\_\_**

1. Find the errors

int nValue = 5;

double dValue = 7.0;

int \*nPtr = &nValue;

double \*dPtr = &dValue;

nPtr = &dValue;

dPtr = &nValue;

1. Tracing the output.

assume the address of nValue is equal to 2333

int main(){

double nValue = 7;

double \*pnPtr = &nValue;

cout << pnPtr << endl;

cout << pnPtr+1 << endl;

cout << pnPtr+2 << endl;

cout << pnPtr+3 << endl;

}

b)

#include <iostream>

using namespace std;

void bill(int x, int \*y)

 {

 cout << "The arguments passed to bill are x ="<< x << " \*y = " << \*y<<endl;

 x = x + 5;

 \*y = \*y + 5;

 cout << "The arguments after being modified are x ="<< x << " \*y = " << \*y<<endl;

 }

void main( )

 {

 int a = 5;

 int b = 10;

 cout << "The values before bill called are a ="<<a<<", b ="<< b<<endl;

 bill (a,&b);

 cout << "The values after bill are a ="<<a<<", b ="<< b<<endl;

 }

c)

#include <iostream>

using namespace std;

int main ()

{

 int numbers[5];

 int \* p;

 p = numbers;

 \*p = 10;

 p++;

 \*p = 20;

 p = &numbers[2];

 \*p = 30;

 p = numbers + 3;

 \*p = 40;

 p = numbers;

 \*(p+4) = 50;

 for (int n=0; n<5; n++)

 cout << numbers[n] << ", ";

 return 0;

}

d)

#include <iostream>

using namespace std;

int main ()

{

 int anArray[5] = { 9, 7, 5, 3, 1 };

cout << \*(anArray+1) << endl;

 system("pause");

 return 0;

}

e)

#include<iostream>

using namespace std;

///////////////////////////////////

class Person {

 // Data members of person

public:

 Person(int x) { cout << "Person::Person(int ) called" << endl; }

};

class Faculty : public Person {

 // data members of Faculty

public:

 Faculty(int x):Person(x) {

 cout<<"Faculty::Faculty(int ) called"<< endl;

 }

};

class Student : public Person {

 // data members of Student

public:

 Student(int x):Person(x) {

 cout<<"Student::Student(int ) called"<< endl;

 }

};

class TA : public Faculty, public Student {

public:

 TA(int x):Student(x), Faculty(x) {

 cout<<"TA::TA(int ) called"<< endl;

 }

};

 void main() {

 TA ta1(30);

}